

- [89] S. Cao, N. He, Y. Guo, and H. Wang, “WASMixer: Binary Obfuscation for WebAssembly,” *arXiv e-prints*, p. arXiv:2308.03123, Aug. 2023.
- [90] M. Jacob, M. H. Jakubowski, P. Naldurg, C. W. N. Saw, and R. Venkatesan, “The superdiversifier: Peephole individualization for software protection,” in *International Workshop on Security*, pp. 100–120, Springer, 2008.
- [91] M. Henry, “Superoptimizer: a look at the smallest program,” *ACM SIGARCH Computer Architecture News*, vol. 15, pp. 122–126, Nov 1987.
- [92] V. Le, M. Afshari, and Z. Su, “Compiler validation via equivalence modulo inputs,” in *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI ’14, p. 216–226, 2014.
- [93] B. Churchill, O. Padon, R. Sharma, and A. Aiken, “Semantic program alignment for equivalence checking,” in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, (New York, NY, USA), p. 1027–1040, Association for Computing Machinery, 2019.
- [94] N. Harrand, C. Soto-Valero, M. Monperrus, and B. Baudry, “Java decompiler diversity and its application to meta-decompilation,” *Journal of Systems and Software*, vol. 168, p. 110645, 2020.
- [95] M. Zalewski, “American fuzzy lop,” 2017.
- [96] L. de Moura and N. Bjørner, “Z3: An efficient smt solver,” in *Tools and Algorithms for the Construction and Analysis of Systems* (C. R. Ramakrishnan and J. Rehof, eds.), (Berlin, Heidelberg), pp. 337–340, Springer Berlin Heidelberg, 2008.
- [97] P. M. Phothilimthana, A. Thakur, R. Bodik, and D. Dhurjati, “Scaling up superoptimization,” *SIGARCH Comput. Archit. News*, vol. 44, p. 297–310, mar 2016.
- [98] R. El-Khalil and A. D. Keromytis, “Hydan: Hiding information in program binaries,” in *Information and Communications Security* (J. Lopez, S. Qing, and E. Okamoto, eds.), (Berlin, Heidelberg), pp. 187–199, Springer Berlin Heidelberg, 2004.
- [99] V. Singhal, A. A. Pillai, C. Saumya, M. Kulkarni, and A. Machiry, “Cornucopia: A framework for feedback guided generation of binaries,” in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, ASE ’22, (New York, NY, USA), Association for Computing Machinery, 2023.

- [100] B. Cox, D. Evans, A. Filipi, J. Rowanhill, W. Hu, J. Davidson, J. Knight, A. Nguyen-Tuong, and J. Hiser, “N-variant systems: a secretless framework for security through diversity,” in *Proc. of USENIX Security Symposium*, USENIX-SS’06, 2006.
- [101] D. Bruschi, L. Cavallaro, and A. Lanzi, “Diversified process replicas for defeating memory error exploits,” in *Proc. of the Int. Performance, Computing, and Communications Conference*, 2007.
- [102] B. Salamat, A. Gal, T. Jackson, K. Manivannan, G. Wagner, and M. Franz, “Stopping buffer overflow attacks at run-time: Simultaneous multi-variant program execution on a multicore processor,” tech. rep., Technical Report 07-13, School of Information and Computer Sciences, UC Irvine, 2007.
- [103] G. Agosta, A. Barengi, G. Pelosi, and M. Scandale, “The MEET approach: Securing cryptographic embedded software against side channel attacks,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1320–1333, 2015.
- [104] T. Jackson, B. Salamat, A. Homescu, K. Manivannan, G. Wagner, A. Gal, S. Brunthaler, C. Wimmer, and M. Franz, “Compiler-generated software diversity,” in *Moving Target Defense*, pp. 77–98, Springer, 2011.
- [105] A. Amarilli, S. Müller, D. Naccache, D. Page, P. Rauzy, and M. Tunstall, “Can code polymorphism limit information leakage?,” in *IFIP International Workshop on Information Security Theory and Practices*, pp. 1–21, Springer, 2011.
- [106] A. Voulimeneas, D. Song, P. Larsen, M. Franz, and S. Volckaert, “dmvx: Secure and efficient multi-variant execution in a distributed setting,” in *Proceedings of the 14th European Workshop on Systems Security*, pp. 41–47, 2021.
- [107] I. Bow, N. Bete, F. Saqib, W. Che, C. Patel, R. Robucci, C. Chan, and J. Plusquellic, “Side-channel power resistance for encryption algorithms using implementation diversity,” *Cryptography*, vol. 4, no. 2, 2020.
- [108] R. L. Castro, C. Schmitt, and G. D. Rodosek, “Armed: How automatic malware modifications can evade static detection?,” in *2019 5th International Conference on Information Management (ICIM)*, pp. 20–27, 2019.
- [109] R. Sasnauskas, Y. Chen, P. Collingbourne, J. Ketema, G. Lup, J. Taneja, and J. Regehr, “Souper: A Synthesizing Superoptimizer,” *arXiv preprint 1711.04422*, 2017.
- [110] J. Cabrera Arteaga, P. Laperdrix, M. Monperrus, and B. Baudry, “Multi-Variant Execution at the Edge,” *arXiv e-prints*, p. arXiv:2108.08125, Aug. 2021.